REMARKS

Claims 1-41 are pending. Claims 9, 15 and 26 were amended solely to correct grammatical errors. New dependent claims 34-41 were added to further define the present invention. No new matter was added since "application programs" are referred to throughout the specification.

Claims 1-33 correspond to claims 28-60 of the parent Application No. 09/873,351, now abandoned. In the last Office Action of the parent application, claims 28-60 were rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Weiss. For at least the reasons set forth below, re-presented claims 1-33 are believed to be patentable over Weiss.

1. Overview of Weiss

Weiss describes a system for enhancing the security of a "secure token code" which is obtained from a token. A secure token code is defined in Weiss as follows:

> a long code, typically several hundred to several thousand bits in length, but sometimes substantially less (for example 64 bits) which is nearly always secret and may be stored in some form in a token; examples include encryption key, private key, biocharacteristic template data, data, confidential information or the like. (column 1, lines 45-51)

A token is defined in Weiss as follows:

> a device which is usually portable and/or personal (but is not limited to being either) and which stores machine and/or visually readable data which is usually secret; nonlimiting examples of tokens are a credit card, smart card, photo-optical data storage card, floppy disk, touch memory button, data key, processor memory component (i.e. RAM, ROM, electronically alterable memory), other data-containing electronic component, other data-containing IC chip, or the like. (column 1, lines 30-38)

Fig. 2A discloses a token 30 in the form of a "smart card." The smart card token contains a processor 40 which can be used to generate the secure token code, and which is then stored in the token 30 (column 4, lines 45). The processor 40 may also include software for using an

encryption key/private key to operate on inputted data to perform encryption/decryption thereon (column 4, lines 61-64).

Weiss further discloses a plurality of terminals 12A-12N, each including a respective processor 16A-16N. Weiss further describes the relationship of the token 30 and the terminal 12/processor 16 as follows:

> More specifically, what is stored in the token is a multibit token secret or sequence which, when algorithmically combined in a particular way with the secret code of the authorized user, produces the secure token code. The algorithmic combination to generate the secure token code may either be performed in the token, and the secure token code temporarily stored therein, or such calculations may be performed at a terminal or processor, with the user inputting his secret code to the terminal or processor and a token reader at the terminal or processor reading the multibit token secret from the token. (column 4, lines 7-16)

Figs. 3-6 are flow diagrams of various processes for performing the above-identified functions. The processes shown in Figs. 3 and 5 are performed in the external terminal/processor 12/16, and at best, receive only inputs from a token. The processes shown in Figs. 4 and 6 are performed in the token and use the token processor 40. A mixed embodiment is described on column 13, line 44 through column 14, line 7 which uses the token processor 40 and the terminal's processor 16.

2. Weiss does not disclose or suggest the structure or organization of its software for implementing its functions

Even though Weiss discloses various processes that allow a token processor 40 to interact with a terminal processor 16, Weiss has no disclosure whatsoever regarding the structure or organization of the software for implementing the processes. Most importantly, Weiss provides no details as to whether the software that executes in the token processor 40 is part of the software that executes in the terminal processor 16, or whether each processor has separate software programs that can interact with each other when appropriate. There are many different ways to implement the functions specified in the flow diagrams of Figs. 3-6, and in the mixed embodiment described in column 13, line 44 through column 14, line 7 which uses the token processor 40 and the terminal's processor 16. All that Weiss says about the software is that the

processor in the token may include software means for utilizing an encryption key/private key to operate on inputted data to perform encryption/decryption thereon (see, column 4, lines 61-64), and that "[e]ach processor 16 would contain suitable storage elements and other suitable hardware and software" (column 7, lines 4-6). These text portions provide absolutely no guidance to an artisan as to how to structure or organize the software. It is hindsight reconstruction of Applicants' invention to presume that the software would be structured and organized in the manner described and claimed by Applicants, particularly since conventional programming techniques are completely contrary to the claimed invention.

Notwithstanding the complete absence of any software programming details in Weiss, the last Office Action of the parent application attempts to match up claimed elements with corresponding portions of Weiss. However, in every instance, the Office Action merely presumes that Weiss has a single software program with a first portion of software code disposed in, and executing on, the terminal's processor 16, and a second portion of software code that includes one or more fragments of code of the software program disposed in, and executing on, the token processor 40. As discussed above, there is no disclosure or suggestion in Weiss of any structure or organization of the software, and thus the element/step matching process set forth in the last Office Action of the parent application is fundamentally flawed.

Furthermore, Figs. 3-6 of Weiss do not appear to even describe multi-processor computer environments. As discussed above, the processes shown in Figs. 3 and 5 are performed in the external terminal/processor 12/16, and at best, receive only input values from a token. The processes shown in Figs. 4 and 6 are performed in the token and use the token processor 40. At best, only the embodiment is described on column 13, line 44 through column 14, line 7 which uses the token processor 40 and the terminal's processor 16 appears to come conceptually close to a multi-processor computer environment. However, even this embodiment has absolutely no disclosure regarding the structure or organization of the software that executes in each of the respective processors.

3. Weiss does not disclose or suggest the concept of identifying and distributing fragments of code of a software program

Since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss identifies or distributes fragments of code of a software program to an external device, or that fragments of such code are executed in an external device.

4. Patentability of claims 1 and 9 over Weiss

Claims 1 and 9 are equivalent to claims 28 and 36, respectively, in the parent application.

Unlike Weiss which merely describes a multi-processor computer environment wherein different programming functions are performed in different processors but says nothing about the structure or organization of the software for performing the functions, claims 1 and 9 explicitly recite the structure and organization of the software code.

Claim 1 recites a multi-processor computer environment wherein each software program includes (i) a first portion of software code to be executed in a computer, and (ii) a second portion of software code that includes one or more fragments of code of the software program, wherein the second portion of code is executed in one or more external devices which are in communication with the computer.

Claim 9 recites a multi-processor computer environment wherein each software program includes (i) a first portion of software code, and (ii) a second portion of software code that includes one or more fragments of code of the software program, and further defines an apparatus comprising a first computer which executes the first portion, and one or more externals unit in communication with the computer, the one or more external units executing the second portion of the software code.

In the last Office Action of the parent application, the Examiner refers to various portions of Weiss as allegedly disclosing the elements and steps in claims 1 and 9. However, as discussed above, it is totally speculative as to whether any software code executing in the token processor 40 constitutes fragments of code associated with the software program that also executes on another computer. For example, there may be one software program associated with the terminal's processor 16 and another software program associated with the token processor 40. This software structure would not meet the claim limitations and would actually teach against

them. Furthermore, the fact that the processes shown in Figs. 3 and 5 are performed in the external terminal/processor 12/16 and the processes shown in Figs. 4 and 6 are performed in the token and use the token processor 40 actually suggests that an artisan implementing Weiss's invention would likely prepare separate software programs for the terminal's processor 16 and the token processor 40 instead of attempting to write a single software program that implements the completely different scheme set forth in the claimed invention. The mixed embodiment described on column 13, line 44 through column 14, line 7 which uses the token processor 40 and the terminal's processor 16 would require coordination between the two programs, but it is well-known in the computer programming field to make two programs interact with each other.

Lastly, since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss executes fragments of code of a software program in the token processor 40.

In sum, there is simply no basis to conclude that the software code in Weiss is structured and organized as set forth in the claimed invention. The fact that it might be capable of being structured and organized as such does not provide a sufficient motivation to rejection claim 1 or 9 over Weiss.

5. Patentability of claim 15 over Weiss

Claim 15 is equivalent to claim 42 in the parent application.

Claim 15 recites a method of transforming a computer program which includes software code. In the method, one or more fragments of the software code are identified. A program call is then associated with each of the identified fragments. The program call is then inserted into the software code, thereby transforming the computer program. When a program call is reached, the respective fragment of software code is executed.

Two commonly accepted definitions of "transform"[1] are (1) to change markedly the appearance or form of, and (2) to change the nature, function, or condition of; convert

[1] The American Heritage® Dictionary of the English Language, Fourth Edition, Copyright © 2000 by Houghton Mifflin Company.

Weiss has no disclosure or suggestion of the claimed transforming process. In the last Office Action of the parent application, the Examiner refers to the mixed embodiment is described on column 13, line 44 through column 14, line 7 which uses the token processor 40 and the terminal's processor 16 as allegedly disclosing such a process. However, this embodiment is described only by reference to selected flowcharts which do not contain any details regarding the structure or organization of the software in the two processors, or how the software was created.

First, no method of transforming a computer program is even described in Weiss. Specifically, there is no disclosure or suggestion in Weiss that fragments of software code in a computer program are identified, or that program calls are associated with such fragments of code and inserted into software code, thereby transforming a computer program. Weiss merely describes the steps for "practicing the teaching of the invention" (column 13, lines 44-45), not the steps for modifying an original computer program to create a transformed computer program that can function in the multi-processor environment described in column 13, line 44 through column 14, line 7.

Second, since the structure and organization of the software associated with the token processor 40 and the terminal's processor 16 are not described in Weiss, it is completely speculative as to whether program calls are used in the software.

Third, since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss even identifies fragments of code of a software program which ultimately are executed in the token processor 40.

6. Patentability of claims 17 and 22 over Weiss

Claims 17 and 22 are equivalent to claims 44 and 49, respectively, in the parent application.

Claim 17 recites a method of executing a computer program which includes software code. The software code has a first portion and a second portion. The second portion includes one or more fragments of the software code and a program call associated with each fragment. The method comprises executing the first portion, and executing the associated fragments when a program call in the second portion is reached.

Weiss does not disclose or suggest the steps of claim 17.

First, there is no disclosure or suggestion in Weiss of any structure or organization of the software and thus there is no basis for asserting that there is a computer program having a first portion and a second portion, wherein the second portion includes one or more fragments of the software code. Nor is there any basis for asserting that there are program calls associated with fragments of software code in Weiss.

Second, since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss executes fragments of code of a software program in the token processor 40.

Claim 22 is an apparatus version of claim 19 and thus is patentable over Weiss for at least the same reasons as given above for claim 19.

7. Patentability of claim 26 over Weiss

Claim 26 is equivalent to claim 53 in the parent application.

Claim 26 recites a method of access control of software code which is executed on a smart card that is in communication with a host computer. The smart card has stored therein access control parameters for identified software code. The host computer uploads software code and its identity data to the smart card. The smart card then uses the access control parameters and the identity data to determine whether access is permissible for the uploaded software code. The software may be executed only if access is permissible.

First, there is no disclosure in Weiss that a "host computer upload[s] software code to [a] smart card" as explicitly recited in claim 26. Weiss never discusses whether the software in the smart card's processor 40 is uploaded from the terminal's processor 16 or an equivalent host-like computer. In the last Office Action of the parent application, the Examiner referred to step 100 of Fig. 5 as allegedly disclosing the uploading step. However, Fig. 5 merely refers to receiving a secret code input for a second token. A secret code input is merely a value and is not software code. Also, Fig. 5 refers to an embodiment wherein data from a token (e.g., smart card) is output to the external processor 16, which is exactly the opposite of the claimed scheme wherein data is uploaded into a smart card.

Second, Weiss has no disclosure or suggestion of the claimed feature of "access control

parameters for...software code...wherein the software code may be executed only if access is permissible." Whether the software in Weiss's token processor 40 executes or not does not depend upon access control parameters. The last Office Action asserts that column 10, lines 41-44 of Weiss discloses the claimed feature of executing software only if access is permissible. However, this portion of Weiss merely describes the Fig. 4 process wherein the algorithmic combining process goes forward only if a private key is stored in the token and a unique input has been generated in the token. These are merely conditions associated with the algorithmic process and have nothing to do with whether access is permissible to execute the software associated with the algorithmic process.

8. Patentability of claim 30 over Weiss

Claim 30 is equivalent to claim 57 in the parent application.

Claim 30 recites a method of executing a plurality of software code fragments of a software program on an external unit, wherein at execution time of each of the software code fragments, the respective software code fragments are automatically uploaded to a memory of the external unit.

Weiss has no disclosure or suggestion that the software contained in the token processor 40 is uploaded only at execution time. In the last Office Action, the Examiner asserts that column 8, lines 24-26 of Weiss discloses this feature. This portion of Weiss reads as follows:

> Before, after or during the reading of the token, the user also inputs his secret PIN or other secret code using character input device 18A (step 64).

This text portion refers only to the inputting of the secret PIN or other secret code. Inputting a secret PIN or the like has nothing whatsoever to do with uploading software code fragments of a software program. A secret PIN is just an alphanumeric sequence. It is not a fragment of software code of a software program.

Furthermore, since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss executes fragments of code of a software program in the token processor 40.

9. Patentability of claim 32 over Weiss

Claim 32 is equivalent to claim 59 in the parent application.

Claim 32 is patentable for at least the same reasons as set forth above with respect to claims 1 and 9 because it recites a specific structure and organization of the software code.

Furthermore, claim 32 recites that prior to compilation of the software program, only the second portion of source code is encrypted. Weiss has no disclosure or suggestion of this feature. In the last Office Action, the Examiner asserts that the token secret store 46 in Fig. 2B discloses this feature. This is incorrect. The token secret store 46 is merely the storage location for the multibit token secret. This element is defined in column 1, lines 52-57 as follows:

> an algorithmically modified secure token code which may be stored in some form in a token and which has no value by itself, but from which the secure token code may be algorithmically derived.

The secure token code is not source code and has nothing whatsoever to do with encrypting any source code stored in the token processor 40. In fact, Weiss says nothing about whether the source code stored in the token processor 40 is encrypted or not.

Lastly, since Weiss has no disclosure of either the structure or the organization of the software used in the token processor 40 or the terminal's processor 16, there is no basis for concluding that Weiss executes fragments of code of a software program in the token processor 40.

10. Patentability of dependent claims

The dependent claims are also patentable over Weiss because they incorporate the limitations of their respective allowable base claims, and because they recite additional patentable limitations.

Conclusion

Insofar as the Examiner's rejections in the last Office Action of the parent application were fully addressed, the instant application is in condition for allowance. Issuance of a Notice

of Allowability of all pending claims is therefore earnestly solicited.

Respectively submitted,

Sigurd Sigbjornsen *et al.*

March 9, 2005 By: Clark Jablon
(Date)

CLARK A. JABLON
Registration No. 35,039
AKIN GUMP STRAUSS HAUER & FELD LLP
One Commerce Square
2005 Market Street, Suite 2200
Philadelphia, PA 19103-7013
Telephone: 215-965-1200
Direct Dial: 215-965-1293
Facsimile: 215-965-1210

7360520 v1